

Detailed study of the column-based priority logic readout of *Topmetal-II*- CMOS pixel direct charge sensor

Mangmang An^a, Chufeng Chen^a, Chaosong Gao^a, Mikyung Han^b, Guangming Huang^a, Rong Ji^a, Xiaoting Li^a, Yuan Mei^{b,*}, Hua Pei^a, Quan Sun^c, Xiangming Sun^{a,*}, Kai Wang^a, Le Xiao^a, Ping Yang^a, Wei Zhang^a, Wei Zhou^a

^aPLAC, Key Laboratory of Quark and Lepton Physics (MOE), Central China Normal University, Wuhan, Hubei 430079, China

^bNuclear Science Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA

^cInstitute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China

Abstract

We present the detailed study of the digital readout of *Topmetal-II*- CMOS pixel direct charge sensor. *Topmetal-II* is an integrated sensor with an array of 72×72 pixels each capable of directly collecting external charge through exposed metal electrodes in the topmost metal layer. In addition to the time-shared multiplexing readout of the analog output from Charge Sensitive Amplifiers in each pixel, hits are also generated through comparators with individually DAC settable thresholds in each pixel. The hits are read out via a column-based priority logic structure, retaining both hit location and time information. The in-array column-based priority logic is fully combinational hence there is no clock distributed in the pixel array. Sequential logic and clock are placed on the peripheral of the array. We studied the detailed working behavior and performance of this readout, and demonstrated its potential in imaging applications.

Keywords: Topmetal, Pixel, Charge sensor, Column-based priority logic, Readout

1. Introduction

Highly pixelated sensors such as CMOS image sensors and Monolithic Active Pixels Sensors (MAPS), have been successfully deployed in various fields. In many nuclear/particle physics applications, the traditional “rolling shutter” style readout (time-shared multiplexing) is orders of magnitude slower than what is required for signal/data acquisition in order to achieve the physics goals. Therefore, many novel readout schemes, designed to access the information collected in pixels in the sensor faster, have been developed over the years. These schemes often exploit the characteristics of signal distribution among pixels, such as the sparseness or clustering in space and time of pixel hits, and the similarity in amplitudes. Notable examples include column-based readout[1–3] and row-based compression [4, 5].

We implemented a column-based priority logic readout in a prototype pixel sensor called *Topmetal-II*- [6], aimed at improving the latency between a pixel hit and the availability of data off the chip. *Topmetal-II*- is implemented in a $0.35 \mu\text{m}$ CMOS process. It features a 72×72 pixel array with $83.2 \mu\text{m}$ pixel pitch. Pixels in *Topmetal-II*- are sensitive to external charges arrived at an exposed metal electrode in the topmost layer in each pixel. Pixel hits are generated by pixel-local comparators with tunable thresholds.

We designed *Topmetal-II*- towards achieving both low analog noise and low latency in digital readout. The analog front-end achieved a $< 14 e^-$ Equivalent Noise Charge (ENC)[6] per pixel. For the digital circuitry, we chose a scheme that is clock-less (fully combinational) in the pixel array to minimize the potential interference from digital activities (flips). Digital activities only happen when the sensor receives hits. The in-array column-based combinational logic drives the address of the pixel that is hit to the edge of the array immediately upon a hit, which minimizes the latency. A sequential logic (with clock) is employed to sense the hit location and time at the edge of the array then ship such information off the sensor.

A detailed study of the analog characteristics of *Topmetal-II*- is reported in [6]. This paper focuses on the details of operation and performance of the digital readout.

2. Sensor structure and operation

A *Topmetal-II*- sensor, as shown in Fig. 1, contains an array of 72×72 sensitive pixels occupying a $6 \times 6 \text{ mm}^2$ area. Each pixel has an exposed metal patch in the topmost layer that can directly collect charge. The charge signal is amplified by a Charge Sensitive Amplifier (CSA) in each pixel (Fig. 2). Light illumination also results in charge signal, which is then amplified by the same CSA. The amplified charge signal is accessible through two channels. The analog voltage signal is read out through a “rolling shutter” style time-shared multiplexer controlled by the array scan

*Corresponding author

Email addresses: ymei@lbl.gov (Yuan Mei),
xmsun@phy.ccnu.edu.cn (Xiangming Sun)

unit. A digital hit signal is generated by an in-pixel comparator with per-pixel adjustable threshold, then read out through the column-based priority logic.

2.1. Column-based priority logic readout

The overall readout has two parts: an in-array combinational logic (Fig. 2, purple dashed box) and a sequential logic (Fig. 2, yellow dashed box) at the bottom edge of the array. The combinational logic consists of a Priority Logic (PL) in each pixel and an Address Bus (AB) in each column. The sequential logic includes a Column Readout Unit (CRU) placed at the bottom edge of each column and a multiplexer (MUX) congregating the outputs of all CRUs. CRUs monitor the address changes on the ABs. CRUs and the MUX are synchronous to shared clock CLK and reset RST signals.

2.1.1. Pixel hits and Priority Logic (PL)

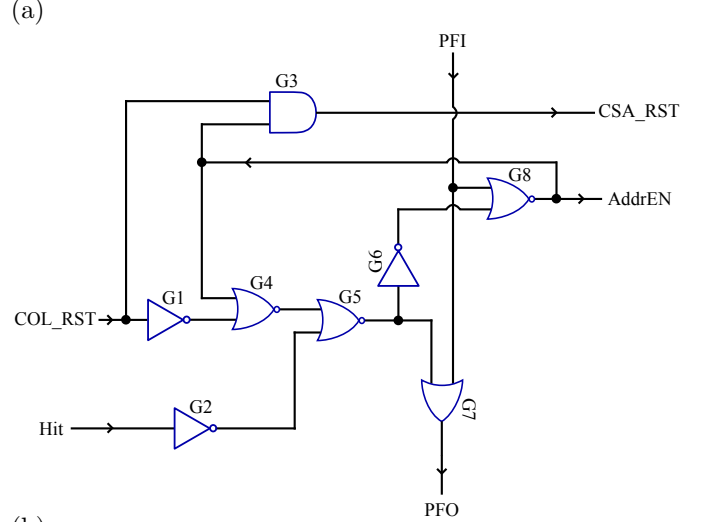
A schematic view of the circuit in a single pixel is shown in the red dashed box in Fig. 2. The exposed *Topmetal* electrode is directly connected to the input of the CSA. A ring electrode (**Gring**), which is in the same topmost metal layer as the *Topmetal*, surrounds the *Topmetal* while being isolated from it. The stray capacitance between the **Gring** and the *Topmetal*, $C_{inj} \approx 5.5$ fF, is a natural test capacitor that allows applied pulses on **Gring** to inject charge into the CSA. The CSA with $C_f \approx 5$ fF converts the injected charge to voltage signal (**CSA_OUT**) and feeds it into the comparator. The comparator compares **CSA_OUT** to a threshold (V_{th}) set by a pixel-local 4-bit DAC (V_{thp}) on top of a common offset V_{thg} that is globally adjustable. $V_{thi} = V_{thpi} + V_{thg}$, where i is the index of pixel in the array. The step size of all the 4-bit DACs is globally adjustable as well. The pixel-local 4-bit DAC is intended for compensating the threshold dispersion of the comparator across the entire array. The CSA and the comparator are constantly active.

Upon an event that **CSA_OUT** surpasses the threshold V_{th} , the comparator asserts **Flag** = 1, which propagates to an AND gate G0 (Fig. 2). The other input of G0, **Mask**, is used for disabling pixels from responding to hits digitally. This feature is exploited during the digital readout tests and imaging demonstration. The **Mask** together with the 4-bits for DAC in each pixel are set by a pixel-local 5-bit SRAM. Writes to SRAMs are synchronous to array scan. SRAMs were chosen over Flip-Flops to save floor space.

When G0 outputs 1, a hit is generated (**Hit** = 1) and the PL module is notified. Each PL is a fully combinational logic that controls the reset (**CSA_RST**) of the CSA upon the readout of a hit and drives the hit information through the column structure. The internal structure of PL and its truth table are shown in Fig. 3.

2.1.2. Column-wise priority chain and Address Bus (AB)

The priority logic signals propagate in columns. For the i th pixel, its **PFI** _{i} is connected to the previous ($(i - 1)$ th) pixel's **PFO** _{$i-1$} , and its **PFO** _{i} is fed into the next ($(i + 1)$ th) pixel's **PFI** _{$i+1$} . Pixels in the same column are daisy-chained in this fashion. Every pixel in the same column has a unique hard-coded 7-bit address in the form of pull-down switches. Encoded pull-down switches are connected to the column-shared Address Bus (AB) (green dashed box in Fig. 2). AB is weakly pulled up to all high by default. When **AddrEN** becomes active in a pixel, said pixel pulls down the AB to its own unique address. The topmost pixel (0th) in a column has **PFI**₀ = 0. If there is no hit in any pixel (**Hit** = 0), the **PFO** output is forced to 0 by G2, G5 & G7, which dictates that every pixel in the column has **PFI** = **PFO** = 0. When there is no active **COL_RST** sent from the CRU module to every pixel in the column simultaneously, the outputs of G3 & G4 are forced to 0. Once a pixel (e.g. i th) gets a hit, due to the effects of G2, G5 & G7, **PFO** _{i} = 1. Forced by G7, all pixels below the i th pixel (denoted by j th, $j > i$) will have **PFI** _{j} = **PFO** _{j} = 1. Forced by G8, any pixel with **PFI** = 1 won't enable **AddrEN** even if it gets a hit. The above described logic forms a column-wise priority chain: only the pixel with a hit that has the lowest i (highest priority) enables its **AddrEN**, and it disables all the pixels lower in the chain from asserting their individual addresses on the AB. Therefore, AB is pulled down by only one pixel (the highest priority pixel with a hit) at any given time so that no race condition rises on the AB.



(b)

Input			AddrEN		Output	
PFI	Hit	COL_RST	Initial	Final	PFO	CSA_RST
0	1	1	0	0	0	0
0	1	1	1	1	1	1
0	1	0	X	1	1	0
0	0	X	X	0	0	0
1	X	X	X	0	1	0

X=do not care.

Figure 3: (a) Schematic of in-pixel Priority Logic (PL) circuitry. (b) Truth table of PL.

1)th) pixel's **PFI** _{$i+1$} . Pixels in the same column are daisy-chained in this fashion. Every pixel in the same column has a unique hard-coded 7-bit address in the form of pull-down switches. Encoded pull-down switches are connected to the column-shared Address Bus (AB) (green dashed box in Fig. 2). AB is weakly pulled up to all high by default. When **AddrEN** becomes active in a pixel, said pixel pulls down the AB to its own unique address. The topmost pixel (0th) in a column has **PFI**₀ = 0. If there is no hit in any pixel (**Hit** = 0), the **PFO** output is forced to 0 by G2, G5 & G7, which dictates that every pixel in the column has **PFI** = **PFO** = 0. When there is no active **COL_RST** sent from the CRU module to every pixel in the column simultaneously, the outputs of G3 & G4 are forced to 0. Once a pixel (e.g. i th) gets a hit, due to the effects of G2, G5 & G7, **PFO** _{i} = 1. Forced by G7, all pixels below the i th pixel (denoted by j th, $j > i$) will have **PFI** _{j} = **PFO** _{j} = 1. Forced by G8, any pixel with **PFI** = 1 won't enable **AddrEN** even if it gets a hit. The above described logic forms a column-wise priority chain: only the pixel with a hit that has the lowest i (highest priority) enables its **AddrEN**, and it disables all the pixels lower in the chain from asserting their individual addresses on the AB. Therefore, AB is pulled down by only one pixel (the highest priority pixel with a hit) at any given time so that no race condition rises on the AB.

2.1.3. Column Readout Unit (CRU)

Each priority chain (column) is terminated by a Column Readout Unit (CRU) at the bottom of the column. CRU monitors the Address Bus (AB) and validates the address change, then records the 7-bit address & 10-bit time stamp for the corresponding hit pixel. Upon the read of a hit, the CRU asserts $\text{COL_RST} = 1$, which is fed back simultaneously to all the pixels in the column. Only the pixel that is pulling on the bus will respond to COL_RST (see Fig. 3), which results in the analog reset of the CSA ($\text{CSA_RST} = 1$), the removal of hit, and the release of the bus. When the bus is successfully released, the address seen by the CRU returns to all high. The CRU senses such condition and outputs $\text{Ready} = 1$. It indicates that a hit has been registered in the column and has not yet been read by the MUX. COL_RST and Ready are kept high until this CRU is read by the MUX. R_en is set to high by the MUX when it reads the associated CRU.

2.1.4. Multiplexer (MUX)

As shown in the blue dashed box of Fig. 2, a digital multiplexer (MUX) polls the status of each CRU sequentially, advancing at the falling edge of each clock cycle. It picks up the valid addresses and time stamps for the hit pixels from each CRU, then ships them off the sensor. A MARKER signal is asserted when the 0th column is polled to indicate the start of a frame. The index of the column being read can be calculated externally referencing to MARKER . A VALID signal is asserted when the column being read has a hit. The address and time stamp outputs are valid only when $\text{VALID} = 1$.

2.2. Readout operation and timing

A timing diagram of the readout process of a valid hit is shown in Fig. 4. It is assumed that there is only one pixel at Row 50, Column 0 is hit and the system counter has an initial value of $\text{Sys_Time}[9:0] = 0001100100_2(100_{10})$. We also set $\text{Mask} = 1$ to enable the pixel response to hits.

Charges arrive at t_1 , causing the CSA output to exceed the threshold of the comparator, resulting in $\text{Flag} = 1$. Since $\text{Mask} = 1$, a hit is generated ($\text{Hit} = 1$); hence, the single-pole-double-throw (SPDT) switch (Fig. 2) grounds the gate of Mf from its original bias FB_VREF so the CSA maximally retains the charge signal. As $\text{PFI} = 0$, following the logic in Fig. 3, PFO and AddrEN become 1 accordingly. At this moment (t_1), the Address Bus (AB) is pulled to the address of this pixel as well ($\text{Addr_Bus} = 0110010_2(50_{10})$). At t_2 (rising edge of the clock in the CRU), the CRU senses the address change and outputs $\text{Addr}[6:0] = 0110010_2(50_{10})$, and waits for 4 clock cycles to confirm that the address change is not a transient phenomenon. At the end of the waiting period, t_3 , the CRU latches the address value and the time stamp from the system counter $\text{Time}[9:0] = \text{Sys_Time}[9:0] = 0001101001_2(105_{10})$. It also sends a reset signal $\text{COL_RST} = 1$ back to the column. Although COL_RST is sent to every pixel in the column, forced by G3 in Fig. 3 (a), only

the pixel that is pulling the AB and is being read out will respond to the reset. The reset sets $\text{CSA_RST} = 1$, which turns on the feedback transistor Mf , discharging C_f so that the CSA output comes down towards the baseline. At t_4 , the CSA output falls below the threshold, causing $\text{Hit} = 0$ hence $\text{AddrEN} = 0$ and $\text{PFO} = 0$. Once $\text{AddrEN} = 0$, CSA reset is done ($\text{CSA_RST} = 0$) and Addr_Bus returns to all high. The CRU also sets $\text{Ready} = 1$ indicating there is a valid hit waiting to be read. Both the COL_RST and Ready are removed when the CRU is polled at t_7 . The time between t_4 and t_7 is non-deterministic and can be as high as 72 clock cycles. During $t_5 \sim t_7$, the MUX is Polling the CRU and shipping the data ($\text{ADDR}[6:0] = 0110010_2(50_{10})$ and $\text{TIME}[9:0] = 0001100100_2(105_{10})$) off the sensor. Since this pixel is in the 0th column, besides generating a $\text{VALID} = 1$, a synchronous MARKER is also simultaneously asserted. As the signal $\text{Polling}(\text{R_en})$ is driven by the falling edge of the clock, it has a half-clock-cycle delay behind the MARKER ; therefore, it's high from t_6 to t_8 .

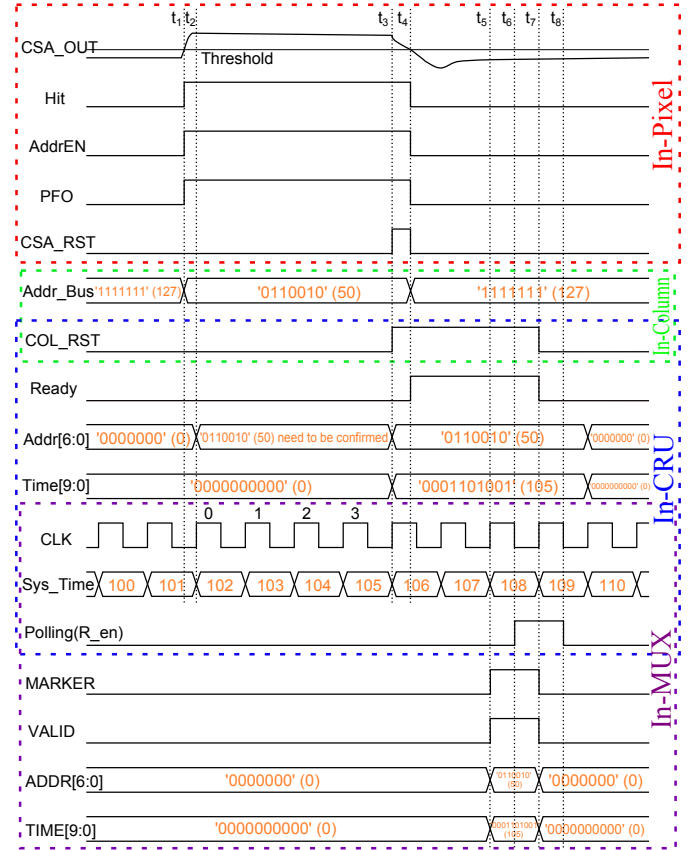


Figure 4: Timing diagram of relevant signal activities during a hit and its readout. In-pixel, in-column, in-CRU and in-MUX signals are indicated in red, green, blue and purple dashed boxes, respectively.

When multiple pixels in the same column are hit simultaneously, the logic reads out and resets the hit pixels sequentially following their priorities in descending order. When a higher-priority hit pixel is waiting to be polled, the CRU keeps the COL_RST high. When $\text{COL_RST} = 1$,

the G1&G4 ensures that the next-priority hit pixel holds its `AddrEN = CSA_RST = 0` until the `COL_RST` is removed. No hit is missed. However, due to this behavior, the CRU cannot respond to the next-priority hit in real-time, which causes the loss of time information for less prior hits. As shown in Fig. 11, the time stamps are only accurate for the pixels with the highest priority.

3. Measurements and experimental results

Controlled signal injections, in the form of test pulses applied on the guard ring (**Gring**) and LED pulsed light illumination, were used to measure the thresholds of every pixel and to demonstrate the imaging capability of the sensor.

3.1. Threshold and noise

We applied a repetitive tail pulse with an amplitude V_{TP} on **Gring** (see the top-left inset in Fig. 2). An equivalent negative charge $Q_i = C_{inj} \times V_{TP}$ is injected at every falling edge of the pulse into the CSA in every pixel simultaneously. The response amplitude of the CSA is expected to be $V_{TP} \cdot (C_{inj}/C_f) \approx 16.5 \text{ mV}$, subject to a small variation due to uncertainties in the capacitance. The CSA responds to both positive and negative charges equally well; however, only the negative equivalent charge can bring the `CSA_OUT` above the threshold to generate hits. Also, we would like to avoid undershoots of the CSA output due to positive charge injections; therefore, tail pulses are chosen over a square wave. The repetition rate of tail pulses is chosen to be low enough such that all the hit pixels have sufficient time to be readout and reset before the next pulse arrives.

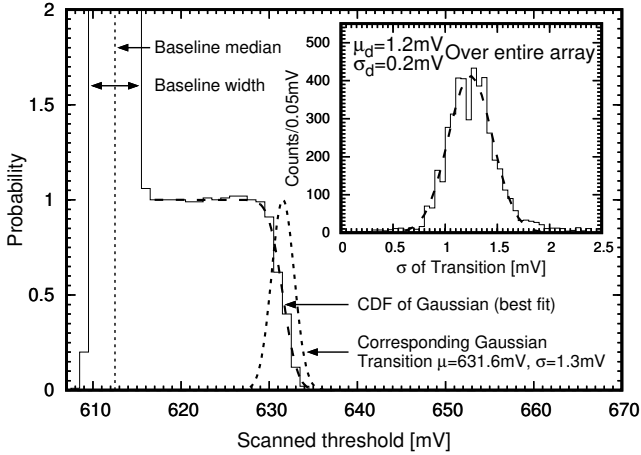


Figure 5: Threshold scan and response parameterization. A representative S-Curve of one pixel is shown. The baseline median and width are determined with the probability > 2 part of the curve (see text for details). The transition region is fitted with the CDF of Gaussian to determine its mean (μ) and width (σ). Inset shows the distribution of the width of the transition (σ) across all the pixels in the array.

As shown in Fig. 5, an S-Curve for a single pixel is obtained by scanning the threshold while recording the corresponding probability for the discriminator and the subsequent logic to register a hit given a test pulse on the **Gring**. The threshold is gradually lowered from well above the signal height where hit probability = 0. When the threshold is close to the injected signal height, a characteristic tapered transition from probability 0 to 1 due to noise appears. When the threshold is close to the baseline, the logic registers a hit every cycle regardless of the injected signal pulse; therefore, the computed probability is bogusly well above 1. When the threshold is well below the baseline, the logic saturates and outputs no hit, although internally the discriminator constantly outputs 1. We determine the median and width of the baseline using the probability > 2 part of the curve. We fit the transition part using the Cumulative Distribution Function (CDF) of Gaussian, $f(x) = \frac{1}{2} \left[1 - \text{erf} \left(\frac{x-\mu}{\sigma\sqrt{2}} \right) \right]$, to determine the mean (μ) and width (σ) of the transition.

The above described procedure is repeated for every pixel in the array. 4-bit DACs are set to 0 for all pixels while the global V_{thg} is varied to achieve the threshold scan. Since the test pulse on **Gring** injects charges into all pixels in the array simultaneously, to avoid unnecessary traffic in the priority chain, we used the **Mask** to enable one row at a time, so that the column readout will read hits from only one pixel in each column. Through the threshold scan procedure for the entire array, we extracted the baseline and transition's location and width from recorded S-Curves of every pixel. The width (σ) of transition, which is an indicator of the noise of CSA output presented to the comparator, has a mean value of 1.2 mV (see the inset in Fig. 5). It is consistent with the analog noise measurement reported in [6]. The baseline median distribution of the array is shown in Fig. 6(a) and (c). The distribution of transition μ -baseline median is shown in Fig. 7. Although the baseline median has a large dispersion due to mismatches in CSA and comparator design, the transition μ -baseline median, which measures the CSA output amplitude response to test pulse injection V_{TP} , remains tightly distributed with a mean value consistent with the expectation $V_{TP} \cdot (C_{inj}/C_f)$.

We write a set of values into the SRAM in each pixel to drive the 4-bit DAC to trim (reduce) the threshold differences between pixels in the array. The set of DAC values, $\{n_i\}$, are calculated from the extracted parameters from threshold scans. The threshold of pixel i is determined by $V_{thi} = V_{thg} + V_{thpi}$. All the 4-bit DACs share a globally adjustable step size V_{step} . $V_{thpi} = V_{step} \times n_i$. Ideally, V_{th} should be as close to the baseline median while kept above the baseline noise width, to detect minimal signal amplitudes. This requirement points to a small V_{step} . However, at the same time, V_{thp} should cover a maximal threshold dispersion of the array in order to reduce the number of dysfunctional pixels due to insufficient trimming. Since n_i has only 16 values, it points to a large V_{step} , contract-

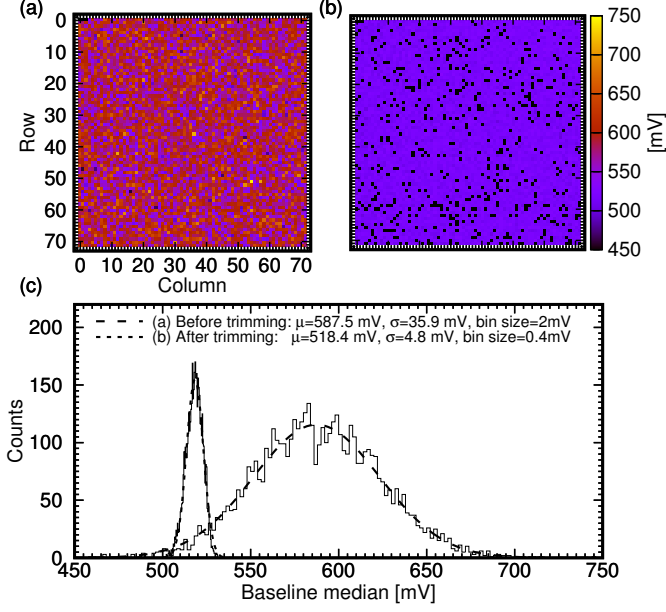


Figure 6: Baseline median distribution across the array in 2D (a, b) and 1D histogram (c) before (a and dashed line in c) and after (b and dotted line in c) trimming with a 4-bit DAC in every pixel. The baseline median is defined in Fig. 6. The reduction of spread (σ) of the baseline median distribution is clearly visible with the trimming. The shift of the mean is unimportant. Black points in (b) indicates the pixels are disabled.

ing the low threshold requirement. To find a balanced set of parameters, we minimize the quadratic sum of signal thresholds, $\sum_i (V_{th} - \text{baseline median})_i^2$, by varying $\{n_i\}$, V_{step} and V_{thg} . We allow a small fraction of pixels with baselines that are far off to be excluded and subsequently disabled. We also disable defective and noisy pixels by setting $Mask = 0$. Disabled pixels are marked with black points in the relevant 2D-figures. A representative set of parameters are $V_{step} = 9 \text{ mV}$, $V_{thg} = 532 \text{ mV}$, and 10% disabled pixels.

After trimming with the optimized setting, we varied V_{thg} to perform the threshold scan again. The results show a greatly reduced width in baseline median distribution (Fig. 6). The signal threshold, however, has a somewhat high mean value and wide distribution (Fig. 7). Ideally, if the trimming were able to equalize all the baselines, which would require an infinitely small V_{step} , the signal threshold distribution would have a width equal to that of the distribution of transition μ –baseline median. A finite (large) V_{step} widens the signal threshold distribution and raises its mean value.

We also extracted the actual step size of the 4-bit DAC in each pixel. The distribution is shown in Fig. 8.

3.2. Imaging with pulsed LED illumination

We placed a purple light LED $\sim 2 \text{ cm}$ above the top surface of a *Topmetal-II* sensor. The sensor is covered by an opaque photo mask with a transparent T-shaped pattern. The T-shaped pattern is aligned with the center of

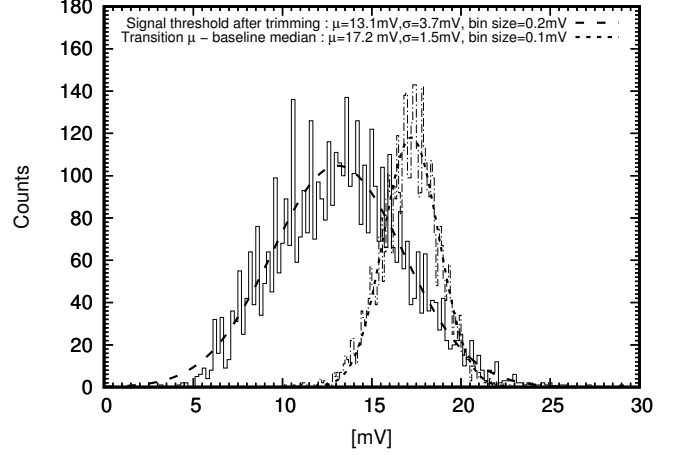


Figure 7: The dotted line shows the distribution of the distance from the transition (μ) to the baseline median of an S-Curve (Fig. 6), which equals to the CSA output amplitude response to the injected pulse height. The dashed line shows the distribution of signal threshold in the array with the optimized trimming settings.

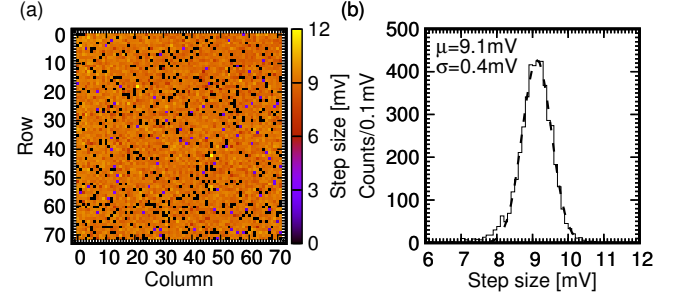


Figure 8: The step size distribution of in-pixel 4-bit DACs in 2D (a) and 1D histogram (b). Black points in the 2D distribution indicates the pixel is faulty and therefore its DAC is not testable.

the sensor (Fig. 9). The LED is driven by a train of narrow pulses with $10 \mu\text{s}$ width and 50 ms interval. The intensity is set such that the illuminated pixels generate hits but their CSAs are not saturated. A $\sim 1 \text{ MHz}$ clock drives the CRUs and the MUX; therefore, the time it takes to read one frame (all 72 columns for once) is $T_f \approx 72 \times 1 \mu\text{s} = 72 \mu\text{s}$. The width of the LED pulse is chosen to be narrow enough to be within one frame. The interval between pulses is large enough to allow all hits to be read out and all pixels to be reset. The sensor operates at the optimized threshold settings.

We recorded many frames of hits induced by a large number of LED pulses. The photo mask was also rotated and displaced to cover different regions of the sensor. Hit location and time are reconstructed from data. A set of images showing the T-shape at four different orientations is in Fig. 10.

When an LED light pulse arrives at the sensor, multiple active pixels that receive the light generate a **Hit** in each of them. Due to the column readout logic, hit pixels that are in the same column will have only one pixel

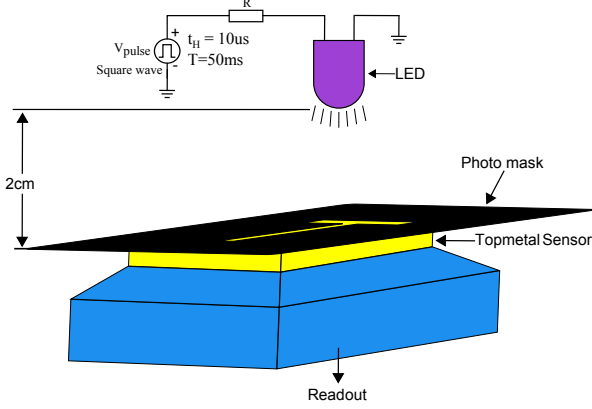


Figure 9: Light pulse injection setup. A violet LED with a peak emission wavelength of 390 nm is placed ~ 2 cm above the top surface of a *Topmetal-II* sensor. Light from the LED is filtered by a photo mask with a T-shaped transparent opening before arriving at the sensor.

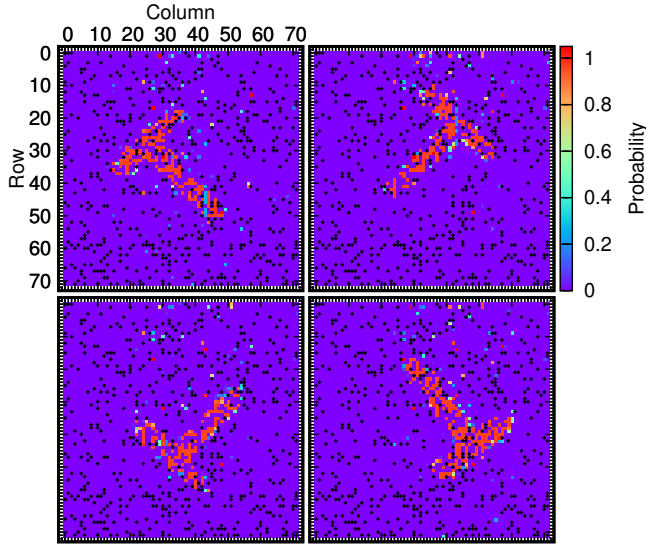


Figure 10: A set of images collected with the T-shaped photo mask placed at four different orientations. Black points mark the disabled pixels.

that has the highest priority registered by the CRU. Since CRUs from each column work concurrently while reading off a single globally shared time, each CRU registers the hit time of the highest priority pixel in its column. Since the light pulse arrives at each pixel simultaneously, the initially registered time, which is from the highest priority pixel, is the same for all the CRUs (Fig. 11 (a)). The MUX reads the registered time from each CRU in a round-robin fashion from one column to the next. When a CRU is read, the pixel of the highest priority in its column is reset, and the CRU subsequently registers the second-highest priority pixel. Since only the CRU has access to the global time, the hit time of the second-highest as well as all the lower priority pixels is determined by the readout rather than the actual arrival of the signal. Only the hit time

of the highest priority pixel is physically meaningful. It is worth noting that starting from the second-highest priority pixel, the time difference between the i th-priority pixel and the $(i + 1)$ th-priority pixel in the same column equals the number of columns (72), which is the time interval between consecutive reads for a given CRU (readout time for one full frame). Fig. 11 (b) exemplifies this phenomenon.

4. Summary and outlook

We successfully implemented a CMOS pixel sensor, *Topmetal-II*, for direct charge collection and imaging. The detailed design, behavior and performance of a column-based priority logic readout in the sensor are presented. The electrical measurements and imaging applications demonstrated the validity of such a readout scheme. The digital readout of pixel hits features a fully combinational logic in the pixel array and a sequential logic in the periphery.

In the current design, although the in-array combinational logic could drive the hit pixel's address to the edge of the array with minimal latency, the sequential logic nature of the CRU and the MUX limits the time it takes to discover the hit information to be beyond one clock cycle. To further reduce the readout latency, analog and combinational logic could be designed at the edge of the array to detect the activities in the Address Bus (AB) promptly. A polling style MUX could be replaced by a priority logic to read out the columns as well. We will investigate these options in future *Topmetal* sensor development in addition to improving the array uniformity.

Acknowledgments

This work is supported, in part, by the Thousand Talents Program at CCNU and by the National Natural Science Foundation of China under Grant No. 11375073. We also acknowledge the support through the Laboratory Directed Research and Development (LDRD) funding from Berkeley Lab, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. We would like to thank Christine Hu-Guo and Nu Xu for fruitful discussions.

References

- [1] J. Millaud, D. Nygren, The column architecture-a novel architecture for event driven 2D pixel imagers, in: Nuclear Science Symposium and Medical Imaging Conference Record, 1995 IEEE, vol. 1, 321-325, 1995.
- [2] P. Yang, G. Aglieri, C. Cavicchioli, P. Chalmet, N. Chanlek, A. Collu, C. Gao, H. Hillemanns, A. Junique, M. Kofarago, M. Keil, T. Kugathasan, D. Kim, J. Kim, A. Lattuca, C. M. Tobon, D. Marras, M. Mager, P. Martinengo, G. Mazza, H. Mugnier, L. Musa, C. Puggioni, J. Rousset, F. Reidt, P. Riedler, W. Snoeys, S. Siddhanta, G. Usai, J. van Hoorne, J. Yi, Low-power priority Address-Encoder and Reset-Decoder data-driven readout for Monolithic Active Pixel Sensors for tracker system, Nuclear Instruments and Methods in Physics Research Section

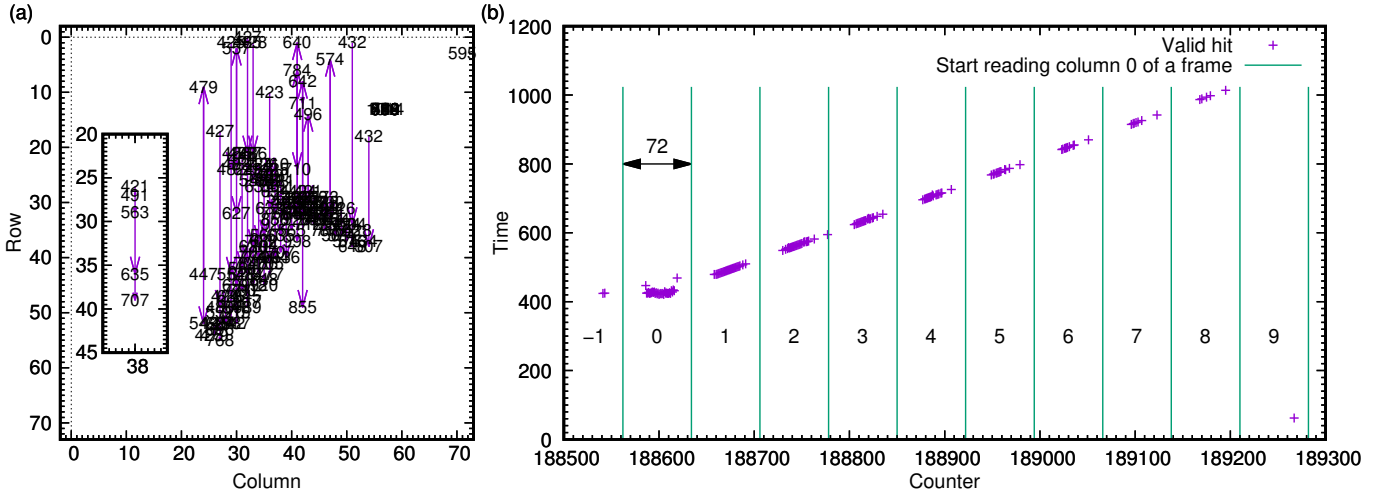


Figure 11: Time stamping of hits. (a) Time stamps printed at the location of corresponding pixels resulting from one light pulse. The T-shaped light hit pattern is clearly visible. Arrows connect hits in the same column, pointing from lower to higher time values. (b) Time stamp (y-axis) read out as a function of clock cycle (Counter, x-axis). Light pulse arrives in section (frame) -1. Frame 0 reads out the initial time stamps from highest priority pixels in each column. Subsequent frames read out pixels hit by the same light pulse but with progressively lower priority. Inset in (a) illustrates the readout of column No. 38 in this fashion.

- A: Accelerators, Spectrometers, Detectors and Associated Equipment 785 (2015) 61 – 69, ISSN 0168-9002, URL <http://www.sciencedirect.com/science/article/pii/S0168900215002818>.
- [3] M. Garcia-Sciveres, D. Arutinov, M. Barbero, R. Beccherle, S. Dube, D. Elledge, J. Fleury, D. Fougeron, F. Gensolen, D. Gnani, V. Gromov, T. Hemperek, M. Karagounis, R. Kluit, A. Kruth, A. Mekkaoui, M. Menouni, J.-D. Schipper, The FE-I4 pixel readout integrated circuit, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 636 (1, Supplement) (2011) S155 – S159, ISSN 0168-9002, URL <http://www.sciencedirect.com/science/article/pii/S0168900210009551>.
- [4] C. Hu-Guo, J. Baudot, G. Bertolone, A. Besson, A. S. Brogna, C. Colledani, G. Claus, R. D. Masi, Y. Degerli, A. Dorokhov, G. Doziere, W. Dulinski, X. Fang, M. Gelin, M. Goffe, F. Guillaux, A. Himmi, K. Jaaskelainen, M. Koziel, F. Morel, F. Orsini, M. Specht, Q. Sun, I. Valin, M. Winter, CMOS pixel sensor development: a fast read-out architecture with integrated zero suppression, Journal of Instrumentation 4 (04) (2009) P04012, URL <http://stacks.iop.org/1748-0221/4/i=04/a=P04012>.
- [5] C. Hu-Guo, J. Baudot, G. Bertolone, A. Besson, A. Brogna, C. Colledani, G. Claus, R. D. Masi, Y. Degerli, A. Dorokhov, G. Doziere, W. Dulinski, X. Fang, M. Gelin, M. Goffe, F. Guillaux, A. Himmi, K. Jaaskelainen, M. Koziel, F. Morel, F. Orsini, M. Specht, Q. Sun, O. Torheim, I. Valin, M. Winter, First reticule size {MAPS} with digital output and integrated zero suppression for the EUDET-JRA1 beam telescope, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 623 (1) (2010) 480 – 482, ISSN 0168-9002, URL <http://www.sciencedirect.com/science/article/pii/S0168900210006078>, 1st International Conference on Technology and Instrumentation in Particle Physics.
- [6] M. An, C. Chen, C. Gao, M. Han, R. Ji, X. Li, Y. Mei, Q. Sun, X. Sun, K. Wang, L. Xiao, P. Yang, W. Zhou, A low-noise {CMOS} pixel direct charge sensor, Topmetal-II-, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 810 (2016) 144 – 150, ISSN 0168-9002, URL <http://www.sciencedirect.com/science/article/pii/S0168900215015727>.